

In the claims:

1. A method for translating from a virtually-addressed bus to a physically-addressed bus, comprising:
 - presenting a virtual address for a memory line on the virtually-addressed bus;
 - 5 initiating snoop processing of an intermediary inclusive storage device coupled to the virtually-addressed bus, the intermediary inclusive device capable of storing information related to the memory line from a main memory coupled to the physically-addressed bus;
 - storing in the intermediary inclusive storage device a pre-fetched memory line including an address tag and data and a pre-fetched status bit, wherein the pre-fetch status bit
 - 10 includes an ON and an OFF indication;
 - switching the pre-fetch status bit to OFF when the virtual address for the pre-fetched memory line is presented on the virtually addressed bus;
 - receiving one of a snoop hit and a snoop miss;
 - if a snoop hit, initiating further snoop processing on local caches coupled to the
 - 15 virtually-addressed bus; and
 - if a snoop miss, accessing a memory location in the main memory.
2. The method of claim 1, wherein when a snoop hit occurs, further comprising reading a coherency bit associated with the memory line, and wherein the status of the coherency bit determines a processes for supplying the memory line in accordance with the presented
- 20 virtual address.
3. The method of claim 1, wherein memory lines are stored in an intermediary inclusive cache.
4. The method of claim 1, wherein address tags are stored in a coherency filter.
5. A method for reducing processing time and bus bandwidth during snoop processing of
- 25 a multi-processor computer architecture, the architecture comprising higher level caches and intermediary caches, the method, comprising:
 - establishing the intermediary caches as inclusive caches, wherein an inclusive intermediary cache includes at least all memory lines of corresponding higher level caches;
 - presenting a virtual address for a memory line on a virtually-addressed bus;
 - 30 initiating snoop processing of the intermediary caches;
 - if receiving a snoop hit, initiating snoop processing on the higher level caches; and
 - if receiving a snoop miss, accessing main memory.

6. The method of claim 5, wherein establishing the intermediary caches as inclusive caches comprises making a capacity of the intermediary caches exceed a total capacity of the corresponding higher level caches.
7. The method of claim 5, wherein establishing the intermediary caches as inclusive
5 caches comprises evicting from any upper level cache a memory line evicted from a corresponding intermediary cache.
8. The method of claim 5, wherein an intermediary cache is implemented as a coherency filter, the method further comprising:
- entering a tag associated with the virtually-addressed memory line into a memory
10 structure of the coherency filter;
 - entering an identity of a processor that owns the memory line; and
 - entering a coherency protocol of the virtually-addressed memory line.
9. A multi-processor computer architecture for reducing processing time and bus bandwidth during snoop processing, comprising:
- 15 a plurality of processors;
 - a plurality of local caches, each local cache corresponding to one of the processors;
 - one or more virtual busses coupled to the local caches and the processors;
 - one or more intermediary caches, wherein at least one intermediary cache is coupled to each virtual bus, each intermediary cache comprising:
- 20 a memory array, and
 - means for ensuring the intermediary cache is inclusive of associated local caches; and
 - a main memory having a plurality of memory lines accessible by the processors.
10. The architecture of claim 9, wherein the ensuring means comprises a capacity of the
25 intermediary cache equal to or greater than a combined capacity of the associated local caches.
11. The architecture of claim 9, wherein the ensuring means comprises a protocol that evicts from any local cache, a memory line evicted from a corresponding intermediary cache.
12. The architecture of claim 9, wherein the memory array is structured to store one or
30 more pre-fetch memory lines, each pre-fetch memory line including:
- an address tag;
 - virtual address bits; and

a pre-fetch status bit, wherein the pre-fetch status bit indicates when a virtual address for the pre-fetch memory line is presented on a virtual bus.

13. The architecture of claim 9, wherein one of the intermediary caches is a coherency filter.

5 14. The architecture of claim 9, wherein one of the intermediary caches is a shared cache.

15. The architecture of claim 9, further comprising a hierarchy of local caches and intermediary caches.

16. The architecture of claim 9, further comprising a physical interconnect coupled to each of the intermediary caches.

10 17. The architecture of claim 16, wherein the physical interconnect is a cross-bar connection.

18. The architecture of claim 16, wherein the physical interconnect is a point-to-point link.

15 19. A mechanism for translating from a virtual bus to a physical interconnect, comprising:
a main memory storing memory lines;
processors coupled to the main memory and capable of accessing the memory lines;
and

means for reducing processing time and bus bandwidth during snoop processing by the processors.

20 20. The mechanism of claim 19, wherein the reducing means comprises one or more inclusive cache means coupled to the physical interconnect and to virtual buses, the virtual buses coupled to the processors.

21. The mechanism of claim 20, wherein the inclusive cache means comprises a capacity of the intermediary cache equal to or greater than a combined capacity of the associated local
25 caches

22. The mechanism of claim 20, wherein the inclusive cache means comprises a protocol that evicts from any local cache, a memory line evicted from a corresponding intermediary cache.